

### 発明の名称

ゲームのキャラクタに画像を投影するプログラム、そのプログラムが組み込まれたゲーム機、及びそのプログラムが格納された記録媒体

5

### 発明の背景

#### 発明の分野

本発明は、ゲームのキャラクタに所望の画面を自由に貼り付ける投影のプログラム、ゲームのキャラクタに画像を投影するプログラムが組み込まれたゲーム機、及びゲームのキャラクタに画像を投影するプログラムが格納された記録媒体に関する。

10

### 関連技術

従来から、ゲームのキャラクタに対してユーザーが所望の画面を貼り付けたり、キャラクタ自身の色や表示などを変えたりすることが行われている。

15 例えば、ゲーム内でユーザーが使用するロボットなどのキャラクタにエンブレムと呼ばれる表示（例えば旗や動物などのイメージ）を所定の位置に貼り付けたりする技術がある。さらに、ユーザーがゲーム内で使用する車のキャラクタに色や模様を付したりする技術もある。ユーザー自身の使用するキャラクタに愛着を持たせるためである。

20 これらは、例えばゲームの開始前に、メインメニューでの所定の選択によりエンブレムの選択画面やエンブレム作成画面、あるいは、色や模様を付したりするための作成画面が選択され、その画面内で作成、選択が行われる。そして、作成後のキャラクタをゲーム機内のメモリなどで保持しておき、ゲーム開始とともに呼び出すことで、ユーザー自身で作成或いは選択したエンブレム等付きのキャラクタが表示され、実際にゲームが開始されることになる。

25

一方で、いわゆる3次元コンピュータグラフィックスで3次元モデルをレ

ンダリングして2次元画像データを映り込ませることも行われている（例えば特許文献1）。

特許文献1 特許公開公報 特開2001-351121号

しかし、従来では、エンブレムのキャラクタに対する貼り付け位置が固定  
5 されており、自身の作成等したエンブレムをキャラクタの様々な場所に自由に貼り付けることができなかった。

キャラクタが固定された位置ではなく様々な位置にエンブレムを自由に貼り付けを行えるようにするために、例えば、固定された位置をそのまま移動させることも考えられる。しかし、移動した位置で、キャラクタの形状が異  
10 なりそのエンブレムがキャラクタに貼り付けられたように見せるために、さらに計算を行わなければならないという問題があった。キャラクタに対する固定位置と移動した位置とでポリゴン面が異なる場合があるからである。

また、車のキャラクタに対して色や模様を付するような場合、その作成面  
15 面は車のキャラクタが平面に展開された画面において行われるため、その画面内で作成して実際に車のキャラクタに貼り付けると、ユーザーの希望する表示とは若干異なる場合がある。実際に車のキャラクタに貼り付けられた表示を予想して作成画面で模様等を作成する必要があるため、何度もやり直したりするなど複雑な操作が必要でかつ熟練した技能を要するという問題があ  
20 った。

#### 発明の要約

そこで、本発明は上記問題点に鑑みてなされたもので、その目的は、ユーザーが作成した表示を、複雑な操作をせずに、キャラクタが複雑な形でも自由  
25 由にそのキャラクタに貼り付けることのできるプログラム、当該プログラムが格納されたゲーム機、及び当該プログラムが格納された記録媒体を提供することにある。

上記目的を達成するために、本発明のプログラムは、画面内で所定の操作を行うための操作手段と、所定の演算を行う演算処理手段と、操作手段と演算処理手段とに接続され演算処理手段を制御する制御手段とを有するゲーム機において、ゲームのキャラクタに所定の画像を投影するプログラムであつて、操作手段の操作により2次元座標からなる画像を制御手段により生成する画像作成ステップと、操作手段からの入力信号に基づいて、画像作成ステップで作成した画像をキャラクタに投影するための仮想光源と、画像とを、三次元仮想空間内においてキャラクタ周囲の任意の位置に配置し、投影光源からキャラクタに画像を投影した投影像をキャラクタに貼付するステップとを  
5 実現させている。

さらに、本発明のかかるキャラクタは、複数のパーツの組み合わせによって構成されており、制御手段に、操作手段の操作に応じてパーツのうち少なくとも一つが画像の投影対象として指定させ、指定されたパーツに対して投影画像を貼り付けさせる、ことを実現させる。

また、上記目的を達成するために、本発明は、画面内で所定の操作を行うための操作手段と、所定の演算を行う演算処理手段と、操作手段と演算処理手段とに接続され演算処理手段を制御する制御手段とを有し、ゲームのキャラクタに所定の画像を投影するプログラムが組み込まれたゲーム機であつて、操作手段の操作により2次元座標からなる画像を前記制御手段により生成する画像作成ステップと、操作手段からの入力信号に基づいて、画像作成ステップで作成した画像をキャラクタに投影するための仮想光源と、画像とを、三次元仮想空間内においてキャラクタ周囲の任意の位置に配置し、投影光源からキャラクタに画像を投影した投影像を前記キャラクタに貼付するステップと、を実現させるためのプログラムが組み込まれたゲーム機にある。  
15 20

さらに、本発明におけるキャラクタは、複数のパーツの組み合わせによって構成されており、制御手段に、操作手段の操作に応じてパーツのうち少なくとも一つが前記画像の投影対象として指定させ、指定されたパーツに対し

て投影画像を貼り付けさせる、ことを実現させるプログラムがゲーム機に組み込まれている。

- また、上記目的を達成するために、本発明は、画面内で所定の操作を行うための操作手段と、所定の演算を行う演算処理手段と、操作手段と演算処理手段とに接続され演算処理手段を制御する制御手段とを有するゲーム機に対してゲームのキャラクタに所定の画像を投影するプログラムが格納された記録媒体であって、操作手段の操作により２次元座標からなる画像を制御手段により生成する画像作成ステップと、操作手段からの入力信号に基づいて、画像作成ステップで作成した画像をキャラクタに投影するための仮想光源と、
- 5 画像とを、三次元仮想空間内においてキャラクタ周囲の任意の位置に配置し、投影光源からキャラクタに画像を投影した投影画像をキャラクタに貼付するステップと、を実現させるためのプログラムが格納された記録媒体にある。

- さらに、本発明は、かかる投影画像をキャラクタに貼付するステップは、キャラクタを構成する各ポリゴンの $\alpha$ 値を所定の値として演算処理手段で演算
- 15 されるよう制御手段を制御する、ことを実現させるためのプログラムが格納された記録媒体としている。

- さらに、本発明は、かかるキャラクタは、複数のパーツの組み合わせによって構成されており、前記制御手段に、前記操作手段の操作に応じて前記パーツのうち少なくとも一つが前記画像の投影対象として指定させ、前記指定
- 20 されたパーツに対して前記投影画像を貼り付けさせることを実現させるためプログラムが格納された記録媒体にある。

- 上記目的を達成するために、本発明は、三次元仮想空間内において、三次元座標からなるオブジェクトに所定の投影画像を投影したオブジェクト画像をディスプレイ装置に表示するための画像データを生成する画像データ生成
- 25 手段としてコンピュータを動作させる画像表示制御プログラムであって、画像表示制御プログラムは、制御手段に、オブジェクトに投影される所定の投影画像データを生成させ、仮想空間内にオブジェクトおよび投影画像を配置

させ、操作者の操作に基づいてオブジェクトに対する投影画像の相対位置と、投影画像をオブジェクトに投影する仮想光源の位置を決定させ、仮想光源とオブジェクト上の投影位置を含む投影面の距離を演算し、仮想光源から所定距離以上はなれた投影面を投影対象から除いて、オブジェクトにおいて前記

5 投影画像を投影すべき投影面を演算させ、仮想光源を視点として投影画像を投影面に投影して、投影画像をオブジェクトの前記投影面に貼り付けさせ、投影画像が貼り付けられたオブジェクトについてのオブジェクト画像データを生成させる機能を実現させることを特徴とする画像表示制御プログラムにある。

10 さらに、本発明は、かかる制御手段に、さらに投影画像の画像データをユーザーが操作手段から入力する操作信号に基づいて生成させる機能を実現させることを特徴とする画像表示制御プログラムにある。

さらに、本発明は、かかる制御手段に、投影画像をオブジェクトと同一ないし近似する形状を備える透明オブジェクトに対して投影して、投影画像を

15 透明オブジェクトに貼り付けさせ、表示装置に投影画像を貼り付けた透明オブジェクトをオブジェクトに重ねて表示させる機能を実現させることを特徴とする画像表示制御プログラムにある。

さらに、本発明は、前記仮想光源と前記投影オブジェクトと前記オブジェクトの位置関係によって、オブジェクトに貼り付けられた投影画像の変形が

20 著しいと判断される場合、投影オブジェクトの透明度を変化させる機能を実現させることを特徴とする画像表示制御プログラムにある。

さらに、本発明は、かかるオブジェクトが複数のパーツの組み合わせによって構成されており、制御部に、操作者の操作に応じてパーツのうち少なくとも一つが投影画像の投影対象として指定させ、指定されたパーツに対して

25 投影画像を貼り付けさせる機能を実現させることを特徴とする画像表示制御プログラムにある。

さらに上記目的を達成するために本発明は、三次元仮想空間内において、

三次元座標からなるオブジェクトに所定の投影画像を投影したオブジェクト  
画像をディスプレイ装置に表示するための画像データを生成する画像データ  
生成手段としてコンピュータを動作させる画像表示制御プログラムであって、  
上記画像表示制御プログラムは、制御手段に、オブジェクトに投影される所  
5 定の投影画像データを生成させ、仮想空間内にオブジェクトおよび投影画像  
を配置させ、操作者の操作に基づいてオブジェクトに対する投影画像の相対  
位置と、投影画像をオブジェクトに投影する仮想光源の位置を決定させ、仮  
想光源とオブジェクト上の投影位置を含む投影面の距離を画素単位で演算し、  
仮想光源を視点として投影画像の画素を視点から最も近い投影面上の画素に  
10 投影して、投影画像をオブジェクトの投影面に貼り付けさせ、前記投影画像  
が貼り付けられた前記オブジェクトについてのオブジェクト画像データを生  
成させる機能を実現させることを特徴としている。これにより、例えば、作  
成したエンブレム画像がオブジェクトを構成する各表面、裏面に貼り付けら  
れたときに、視点から最も近い投影面にのみエンブレム画像が貼り付けられ  
15 る。

本発明によれば、ユーザーが作成した表示を、複雑な操作をせずに、キャ  
ラクタが複雑な形でも自由にそのキャラクタに貼り付けることのできるプロ  
グラムを提供することができる。

## 20 図面の簡単な説明

図 1 は、本発明が適用されるゲーム機の内部構成を示す図である。

図 2 は、オブジェクトを描画するための処理を示すフローチャートである。

図 3 は、エンブレムを貼り付ける対象のオブジェクトを画面表示した場合  
の一例を示す。

25 図 4 は、投影の処理を示すフローチャートである。

図 5 は、エンブレム作成画面の一例を示す図である。

図 6 は、投影の概念を説明するための図である。

図 7 は、実際に投影を行う直前の画面の一例を示す図である。

図 8 は、実際に投影を行った後の画面の一例を示す図である

図 9 は、ゲーム内での表示全体の処理を示すフローチャートである。

図 10 は、ゲーム内での表示全体の処理を示すフローチャートである。

5 図 11 は、ゲーム内での表示全体の処理を示すフローチャートである。

図 12 は、ゲームが実行されている場合の画面表示の一例を示す図である。

図 13 A、B ともに、オブジェクト 16 に貼り付けるエンブレムの例を示す。

図 14 は、テクスチャ面 31 の例を示す図である。

10 図 15 A は、テクスチャ面 31 とエンブレムとの関係を示す斜視図であり、  
図 15 B は側面図である。

図 16 A は、テクスチャ面 31 とエンブレムとの関係を示す斜視図であり、  
図 16 B は側面図である。

図 17 は、 $\alpha$  値を有するテクスチャの例を示す図である。

15 図 18 A は、テクスチャ面 31 とエンブレムとの関係を示す斜視図であり、  
図 18 B は側面図である。

図 19 A は、テクスチャ面 31 とエンブレムとの関係を示す斜視図であり、  
図 19 B は側面図である。

図 20 A は、テクスチャ面 31 とエンブレムとの関係を示す斜視図であり、  
20 図 20 B は側面図である。

図 21 A は、テクスチャ面 31 とエンブレムとの関係を示す斜視図であり、  
図 21 B は側面図である。

#### 好ましい実施例の詳細な記述

25 次に本発明の実施の形態について図面を参照して詳細に説明する。

図 1 は、本発明が適用されるゲーム機 50 の内部構成を示す図である。

ゲーム機 50 は、CPU 1 と、操作部 2 と、主記憶装置 3 と、記録媒体 4

と、ジオメトリックプロセッサ 5 と、ローカルバッファ 6 と、ディスプレイ  
プロセッサ 7 と、テクスチャバッファ 8 と、ディスプレイ 9 とから構成され  
る。

CPU 1 は、操作部 2 と主記憶装置 3 と記録媒体 4、およびテクスチャバ  
5 ュッファ 8 に接続される。操作部 2 によるユーザーの各種操作に関する制御信  
号が入力されるとともに、主記憶装置 3 や記録媒体 4 に記憶された各種プロ  
グラムを実行するためのものである。その詳細は後述する。

操作部 2 は、CPU 1 に接続され、ゲーム機 50 とユーザーとのインター  
フェースとしての役割を有し、ユーザーが種々の操作を行うことで、後述す  
10 るエンブレムの作成や、キャラクタの貼り付け、さらにゲーム実行中におけ  
るキャラクタの操作などを行うことができる、そして操作部 2 は、ユーザー  
の所定の操作に対応した制御信号を CPU 1 に出力する。

主記憶装置 3 は、CPU 1 に接続されるとともにテクスチャバッファ 8 に  
接続されている。主記憶装置 3 には、エンブレム表示のためのプログラムな  
15 ど各種プログラムが格納されている。また、CPU 1 で実行されたプログラ  
ムに対する各種データも記憶されることになる。

記録媒体 4 には、ゲームの種類ごとに異なるプログラムが記録されており、  
例えば CD-ROM などの媒体から構成されている。記録媒体 4 は、ゲーム  
機 50 と着脱可能になっており、記録媒体 4 をゲーム機 50 に装着させると、  
20 記録媒体 4 に格納されたプログラムが CPU 1 の制御により、CPU 1 を経  
由して主記憶装置 3 に出力させるようになっている。

ジオメトリックプロセッサ 5 は、CPU 1、ローカルバッファ 6、及びデ  
ィスプレイプロセッサ 7 に接続されている。ジオメトリックプロセッサ 5 は、  
CPU 1 から出力される各種データが入力され、このデータをもとに画面表  
25 示等を行うための演算を行う。演算した結果、演算データは、ディスプレ  
ィプロセッサ 7 に出力される。

ローカルバッファ 6 は、ジオメトリックプロセッサ 5 と接続され、ジオメ



トリックプロセッサ 5 から出力される演算データを必要に応じて記憶する。

ディスプレイプロセッサ 7 は、テクスチャバッファ 8 と接続されとともに、ディスプレイ 9 に接続されている。ジオメトリックプロセッサ 5 で演算した演算データをもとに実際にディスプレイ 9 に表示させるために演算を行い、

- 5 画面データとしてディスプレイ 9 に出力する。また、ディスプレイプロセッサ 7 は、ジオメトリックプロセッサ 5 からの演算データをもとに適宜、テクスチャバッファ 8 に出力する。

- 10 テクスチャバッファ 8 は、ディスプレイプロセッサ 7 から出力されるテクスチャを一時記憶するとともに、ディスプレイ 9 にも接続され、表示画面（フレーム画面）に対する奥行きを示す Z 値も一時記憶する。この Z 値の記憶については後述する。また、テクスチャバッファ 8 は、CPU 1 及び主記憶装置 3 にも接続され、CPU 1 からの制御信号が入力されて主記憶装置 3 から転送されるデータを一時記憶する。

- 15 ディスプレイ 9 は、ディスプレイプロセッサ 7 から出力される画面データをもとに、実際にゲーム画像を表示するためのものである。また、上述の Z 値をテクスチャバッファ 8 に出力する。

- 20 次に図 2 を用いてエンブレムを貼り付ける対象となるキャラクタ（オブジェクト）を表示させる描画の処理について説明する。この描画の処理のためのプログラムは記録媒体 4 に記録され、記録媒体 4 がゲーム機 50 に装着された場合に CPU 1 の制御により読み出され、主記憶装置 3 に転送されて、主記憶装置 3 に記憶されていることが前提となる。もちろん、当該プログラムが予めゲーム機 50 に組み込まれて主記憶装置 3 に予め記憶されている場合でも適用できる。

- 25 まず、オブジェクトの描画についての処理が開始される（ステップ S 10）と、CPU 1 は、カメラの行列を生成する（ステップ S 11）。ここでは、グローバル座標系（x、y、z）の所定の位置にオブジェクト（キャラクタ）が存在している場合に、その座標系の任意の位置に実際にディスプレイ 9

の画面に表示される視点の位置が存在する。この視点の位置に“カメラ”が位置するものとして、そのカメラの位置等を行列式で表す。そして、この行  
列式がカメラの行列なのである。カメラの行列式に含まれるものは、カメラ  
のグローバル座標系の位置座標の他、グローバル座標系の原点からの角度の  
5 値も含まれる。カメラの行列は、例えば4行3列の行列式や、4行4列の行  
列式で表されるが、もちろんその他n行m列の行列式で表すことも可能であ  
る。生成した行列式は、主記憶装置3に出力して主記憶装置3で記憶される。

次いで、CPU1は、描画開始のリクエストをジオメトリックプロセッサ  
5に出力する（ステップS12）。これによりジオメトリックプロセッサ5  
10 は、オブジェクト描画のための処理を開始することになる。本実施例では、  
ディスプレイ9に表示されるゲーム画像の1フレーム単位の時間（1／60  
秒：1秒で60フレーム表示される場合）ごとにこのリクエストを出力する。  
オブジェクトを含む画像の描画は1フレーム単位で行われるからである。

次いで、CPU1は、ステップS11で生成したカメラの行列式を主記憶  
15 装置3から読み出す（ステップS13）。

次いで、CPU1は、モデルの位置の移動のための処理を行う（ステップ  
S14）。すなわち、ステップS13で読み出したカメラの行列式と、オブ  
ジェクト（モデル）の位置とから、カメラの位置からの相対的なオブジェク  
トの位置を演算する処理を行う。具体的には、ともにグローバル座標系で表  
20 されたオブジェクトの位置とカメラの位置とから、カメラの位置を原点とし  
た場合のオブジェクトの位置を演算する処理を行う。これにより、カメラの  
視点からのオブジェクトの相対位置、すなわち、実際にディスプレイ9の画  
面上に表示されるオブジェクトの位置を求めることができる。

次いで、CPU1は、モデルデータの転送を行う（ステップS15）。具  
25 体的には、CPU1は、ステップS14で演算したオブジェクトの相対位置  
を示すモデルデータを主記憶装置3に出力する。主記憶装置3は、このモデ  
ルデータを一時記憶する。

次いで、CPU 1 は、1 フレームに含まれるすべてのオブジェクトに対してオブジェクトの相対位置を演算したか否か判断する（ステップ S 1 6）。まだ、フレーム内に演算していないオブジェクトがあるなら（ステップ S 1 6 で “N o” のとき）、処理は再びステップ S 1 3 に戻り主記憶装置 3 から

5 カメラの行列式を呼び出して上述したカメラからの相対位置を各オブジェクトに対して演算処理を行う。

1 フレームに含まれるすべてのオブジェクトに対する演算が行われた場合（ステップ S 1 6 で “Y e s” の場合）、CPU 1 は、描画終了のリクエストをジオメトリックプロセッサ 5 に出力する（ステップ S 1 7）。そして、

10 CPU 1 は、このリクエストの出力をトリガとして主記憶装置 3 に記憶した各オブジェクトのモデルデータを読み出して、ジオメトリックプロセッサ 5 に出力する。これにより、ジオメトリックプロセッサ 5 は、モデルデータをもとに実際にディスプレイ画面 9 上にオブジェクトを表示するための演算処理が行われる。

15 すなわち、ジオメトリックプロセッサ 5 から出力される演算データをもとに、ディスプレイプロセッサ 7 は、画面表示のための処理を行い、その出力がディスプレイ 9 に出力される。そして、ディスプレイ 9 では画面上でエンブレムを貼り付ける対象のオブジェクトが表示されることになる（ステップ S 1 8）。そして、オブジェクト描画のための処理が終了する（ステップ S

20 1 9）。

なお、オブジェクトの描画処理（ステップ S 1 8）では、ジオメトリックプロセッサ 5 において各オブジェクトを表現するポリゴンの  $\alpha$  値をすべて “0”（または “1”）にして色表現のない透明なポリゴン面から構成されるオブジェクトを表現している。計算量を減らすためである。

25 この描画の処理により、実際にディスプレイ 9 に表示されるオブジェクトの例を図 3 に示す。ディスプレイ画面 1 5 のほぼ中央には、ユーザーがゲーム内に使用するキャラクタ 1 6 が表示される。このキャラクタ 1 6 がエンブ

レムを貼付するキャラクタである。図面ではわかりにくい、実際には透明のキャラクタが表現されている。

画面 15 の下部には、操作方法 17 が表示される。操作方法 17 は、ディスプレイ 9 の画面上における操作と操作部 2 との対応関係を示すもので、操作部 2 の操作によりキャラクタ 16 の位置を変えたりすることができるようになっている。ちなみに、この画面を表示させるには、例えばメインメニューで個人データ等の選択画面を選択することで表示されるようになっている。

次に、図 3 のように表示されたキャラクタに対して所望のエンブレムを投影して貼付する処理について図 4 を参照して説明するが、この処理が開始される前の段階でエンブレムの作成を行うことが前提となっている

エンブレムの作成画面の例を図 5 に示す。ここでエンブレムとは、図 3 に表示されたキャラクタの所望の位置に貼付するイメージである。ユーザーはこの作成画面で、エンブレムを好みの画像にしたり、編集したりすることができる。この作成画面も、ゲームのメインメニューで個人データ等の所定の選択を行うことで表示されるようになっている。

この作成画面 (EMBLEM EDIT) は、エンブレム作成ボタン群 21 と、エンブレム拡大画面 23 と、表示画面 24 と、色選択ボタン群 25 とから主に構成される。

エンブレム作成ボタン群 21 は、エンブレム 28 を画面内で作成するための複数のボタン 210 ~ 221 から構成され、アイコン 27 を所定の操作部 2 の操作により画面 20 上で選択できるようになっている。

ボタン 210 は、自由に線を拡大画面 23 上で描くためのボタンである。アイコン 27 の画面内の操作でボタン 21 をクリックすると拡大画面 23 内で自由に線を描くことができるようになっている。

ボタン 211 は、いわゆる“消しゴム”のためのボタンで、アイコン 27 の操作により拡大画面 23 で描いた線を消すことができる。

ボタン 212 は、ペイントのためのボタンで、このボタンをクリックして

拡大画面 2 3 内でアイコン 2 7 を所定の位置に移動させ、所定の操作を行うことでアイコン 2 7 が位置する所定範囲内で色が付されるようになされている。

- ボタン 2 1 3 は、四角形を描くためのボタンであり、画面内での操作により拡大画面 2 3 内で四角形を描くことができる。

ボタン 2 1 4 は、描いたエンブレム 2 8 をプレビューするためのボタンであり、ボタン 2 1 5 は、作成画面を終了させるためのボタンである。

ボタン 2 1 6 は、ペンの太さを設定するためのボタンで、拡大画面 2 3 で描くエンブレムの線の太さを設定することができる。

- 10 ボタン 2 1 7 は、拡大画面 2 3 内で描いたエンブレム 2 8 をクリアするためのボタンで、ボタン 2 1 8 は直線を描くためのボタンである。

ボタン 2 1 9 は、円を描くためのボタンであり、ボタン 2 2 0 は、エンブレム 2 8 をさらに拡大させるためのボタンである。そして、ボタン 2 2 1 は、いわゆる“U n d o”機能のボタンであり、直前の操作を取り消して元に戻

- 15 するためのボタンである。

拡大画面 2 3 には、エンブレム 2 8 を拡大表示させ、細かな調整を行うことができるようになっている。

表示画面 2 4 は、後述する図 1 0 に示す投影を行う画面で表示されるエンブレムの大きさとほぼ等しい大きさでエンブレム 2 8 が表示される画面であ

- 20 る。ユーザーはこの画面を確認しながら拡大画面 2 3 においてエンブレム 2 8 を作成することができる。

色選択ボタン群 2.5 は、複数の色がボタンとして表示されており、アイコン 2 7 を画面 2 0 で移動させ所望の色を示すボタンを選択することで、拡大画面 2 3 に表示されたエンブレム 2 8 に色を付すことができる。

- 25 このエンブレム作成画面 2 0 は、以下のようにして表示されることになる。すなわち、記録媒体 4 に記録されたエンブレム作成画面を表示するためのプログラムが格納され、これを CPU 1 が読み出し、表示のための所定のデー

タをジオメトリックプロセッサ 5 に出力する。ジオメトリックプロセッサ 5 は、表示のための処理を行い、演算データをディスプレイプロセッサ 7 に出力する。ディスプレイプロセッサ 7 は演算データをもとに表示のためのデータを生成し、ディスプレイ 9 に出力され、実際のエンブレム作成画面がディスプレイ 9 に表示されることになる。

そして、操作部 2 の所定の操作により、その制御信号が CPU 1 に出力されその制御データをもとに CPU 1 は、再びその操作に対応した表示を上述の動作により行う。終了ボタン 215 の操作によりそれまで作成画面 20 で作成したエンブレム 28、すなわち、ジオメトリックプロセッサ 5 でエンブレム 28 の表示の演算を行うためのデータは、主記憶装置 3 に記憶されることになる。以上がエンブレム作成のための処理である。

再び図 4 に戻って、上述により作成したエンブレム 28 をキャラクタ 16 に貼付する投影の処理について説明する。かかる投影処理のためのプログラムは記録媒体 4 に記録され、記録媒体 4 をゲーム機 50 に装着させた場合に CPU 1 の制御により読み出され、主記憶装置 3 に転送され、この主記憶装置 3 で記憶される。もちろん、当該プログラムが記録媒体 4 ではなく、主記憶装置 3 に予め記憶されており、当該プログラムが組み込まれた組み込み型のゲーム機 50 でも適用される。

投影の処理が開始される（ステップ S30）と、CPU 1 はライト行列の生成を行う。ライト行列の生成とは、グローバル座標系で表示されたキャラクタ 16 に対して、任意の位置（ライト位置）を示す行列式を生成することである。このライト位置からの視点がエンブレム 28 をキャラクタ 16 に貼り付けを行うための画面の視点となる。ちなみに、ここではライトの位置と上述したカメラの位置とは同じである。位置が異なる場合とは、例えば実際にゲームを実行している最中にキャラクタ 16 自身はカメラからの視点、エンブレム 28 はライトからの視点とする場合が考えられる。このようにすることで、エンブレム 28 のみが例えばキャラクタ 16 の側面に実際の画面か

らの視点とは異なる視点のエンブレム 28 を表示することができるようになる。

図 6 には、グローバル座標系で所定の位置にキャラクタ 16 とライトの位置 30 とが存在する図を示す。この場合ライトの位置 30 は、キャラクタ 16 に対して右側後方に位置している。このライトの位置 30 とカメラの位置とが同じ場合には、ライトの位置 30 からの視点が実際の画面上に表示されることになる。

さらに、図 6 に示すようにライトの位置 30 から所定の距離離れた位置にエンブレムテクスチャ面 31 が存在する。これは上述したエンブレム作成画面 20 (図 5 参照) で作成したエンブレム 28 が表示されるテクスチャ面である。

ライトの位置とカメラの位置とはこの場合同じであるので、ライト行列は上述したカメラの行列と同様の行列式で表現される。作成したライト行列は主記憶装置 3 に出力し、一時記憶される。

再び図 4 に戻って、ライト行列を生成した CPU 1 は、次いで、投影行列を生成する (ステップ S 32)。この投影行列は、図 6 に示すエンブレムテクスチャ面 31 が表示される位置や、テクスチャ面 31 の大きさを示す行列式である。ここでエンブレムテクスチャ面 31 の位置は、グローバル座標系ではライトの位置 30 から一定の距離に位置しているため、この距離やテクスチャ面 31 の大きさからこの投影行列を作成することができる。エンブレムテクスチャ面 31 は、初期設定で例えば  $0.5 \times 0.5$  で表される。 $0.5$  はディスプレイ 9 に表示される画面の半分であり、 $0.5 \times 0.5$  は画面の縦、横それぞれ半分の大きさを示す。生成した投影行列は、主記憶装置 3 に出力し、一時記憶される。

次いで、CPU 1 は、ビュー行列の生成を行う (ステップ S 33)。このビュー行列の生成は、ステップ S 31 で生成したライト行列とステップ S 32 で生成した投影行列とを乗算することで生成できる。すなわち、各ステッ

ブで主記憶装置 3 に記憶されたライト行列と投影行列とを CPU 1 が読み出し、乗算することで生成される。

- 次いで、CPU 1 は、カメラの行列を生成する（ステップ S 3 4）。上述したようにカメラの行列は、ディスプレイ 9 の画面上の視点を示す行列式で、
- 5   ここではライトの位置 3 0 とカメラの位置とは一致しているので、ライト行列と同内容の行列式を生成することになる。生成の具体的処理は、上述した図 2 のオブジェクトの描画処理におけるカメラの行列の生成（ステップ S 1 1）と同様である。

- 次いで、CPU 1 は、ジオメトリックプロセッサ 5 に描画開始のリクエスト
- 10   トを出力する（ステップ S 3 5）。このリクエストを出力するタイミングも上述したオブジェクトの描画処理（図 2）と同様に、1 秒で 6 0 フレーム表示できる本実施例のゲーム機 5 0 では 1 / 6 0 秒ごと（1 フレーム表示される時間ごと）に出力する。

- 次いで、CPU 1 は、エンブレムテクスチャをテクスチャバッファ 8 に出力する（ステップ S 3 6）。図 5 で示すエンブレム 2 8 は、主記憶装置 3 に
- 15   記憶されているので、CPU 1 は、主記憶装置 3 からエンブレム 2 8 の表示を行うためのデータを読み出し、テクスチャバッファ 8 に出力し、テクスチャバッファ 8 に記憶される。

- 次いで、CPU 1 は、テクスチャ座標の生成を行う（ステップ S 3 7）。
- 20   すなわち、作成したエンブレム 2 8 がキャラクタ 1 6 の各ポリゴン面に貼付されるように、テクスチャ面 3 1 に位置するエンブレムの座標をキャラクタ 1 6 の損算する位置に変換する。このテクスチャ座標の生成は、ステップ S 3 3 で生成したビュー行列とキャラクタ 1 6 の 3 次元座標とを用いて生成する。エンブレム 2 8 はテクスチャ面では、2 次元座標（いわゆる u-v 座標系）であるが、キャラクタ 1 6 のポリゴンを形成する各頂点座標は 3 次元座標であるため、生成するテクスチャ座標は 3 次元座標として表現される。か
- 25   かる、テクスチャ座標は、キャラクタ 1 6 上の座標に変換された座標である



ので、ユーザーの作成したエンブレムがキャラクタ 16 上に貼付された画像が表示できるようになる。

次いで、CPU 1 は、カメラの行列をロードする（ステップ S 38）。すなわち、CPU 1 は、ステップ S 34 で生成したカメラの行列を主記憶装置

5 3 から読み出す処理を行う。

次いで、CPU 1 は、モデルの位置の移動処理、すなわち、カメラの視点からの相対的なオブジェクト（ここではエンブレム 28 が貼付されたキャラクタ 16）の位置の変換処理を行う（ステップ S 39）。オブジェクトの描画の処理と同様に、カメラの視点がディスプレイ 9 の実際の画面からの視点

10 になり、グローバル座標系では任意の位置にあるカメラの位置に対して、このカメラ位置を例えば原点としてオブジェクトの位置の座標を演算する。

次いで、CPU 1 は、ステップ S 39 で演算した各データを主記憶装置 3 へ出力し（ステップ S 40）、主記憶装置 3 はこれらのデータ（モデルデータ）を記憶する。

15 次いで、1 フレーム内のすべてのオブジェクトに対する相対位置の演算を行うと、描画終了のリクエストをジオメトリックプロセッサ 5 へ出力する（ステップ S 41 で “Y e s” のとき）。描画終了のリクエストを CPU 1 がジオメトリックプロセッサ 5 へ出力しないとき（ステップ S 41 で “N o” のとき）、すなわち、1 フレームに含まれるすべてのオブジェクトに対する  
20 演算を終了していないとき、処理は再びステップ S 35 に移行し、上述の処理を繰り返すことになる。

そして、CPU 1 は、このリクエストの出力をもとにステップ S 40 で主記憶装置 3 に記憶したモデルデータを読み出して、ジオメトリックプロセッサ 5 へ出力する。これにより、ジオメトリックプロセッサ 5 は、実際にディ  
25 スプレイ画面 9 上にエンブレムが貼付されたオブジェクトを表示するための処理が行われる。さらに、ジオメトリックプロセッサ 5 から出力される演算データからディスプレイプロセッサ 7 は、画面表示のための処理を行い、そ

の出力がディスプレイ 9 に出力される。そして、ディスプレイ 9 では画面上でエンブレムが貼付されたオブジェクトが表示されることになる（ステップ S 4 2）。そして、オブジェクト描画のための処理が終了する（ステップ S 4 3）。

- 5 この投影の処理により実際にディスプレイ 9 上に表示される、オブジェクトの例を図 7、図 8 に示す。図 7 は、キャラクタ 1 6 にエンブレムを貼り付ける直前の画面であり、処理としてはテクスチャ座標の生成を行っていない（ステップ S 3 7 の処理が行なわれていない場合）もので、図 8 はエンブレムをキャラクタ 1 6 に貼り付けた場合（ステップ S 3 7 でテクスチャ座標の生成が行われた場合）の画面を示す。

- 図 7 は、エンブレム 2 8 がまだキャラクタ 1 6 に貼り付いていないので、エンブレム 2 8 は、図 5 に示すエンブレム作成画面 2 0 の実際の表示画面 2 4 に表示されたエンブレム 2 8 がそのまま画面 1 5 に表示されている。画面 1 5 の下部には操作方法 1 7 が表示され、この記述に基づいてユーザーは操  
15 作部 2 を操作することで、エンブレム 2 8 の位置を移動させたり、その大きさを変化させたりすることができる。

- 操作部 2 によってエンブレムの大きさを変化させたとき、その変化に対応する制御信号が操作部 2 から CPU 1 に出力され、この制御信号に基づいて CPU 1 は、上述の投影の処理（図 4）のステップ S 3 2 の投影行列の生成  
20 で、その大きさに対応した投影行列を生成する。すなわち、大きさが変化すると初期設定の 0. 5 の値が変化してその大きさの値に対応する値を用いて投影行列を生成して上述の投影の処理を行うことになる。

- また、エンブレムの位置を変化させたいときは、操作部 2 によって所定の操作により行われる。すなわち、その変化させた位置に対応する制御信号が  
25 操作部 2 から CPU 1 に出力される。そして CPU 1 は、この制御信号に基づいて画面の視点位置、すなわちグローバル座標系のライトの位置や原点からの角度が変化し、その移動した位置に対応するライト行列をステップ S 3

1 生成して以後上述の処理（図 4 参照）を行って、エンブレム 28 の移動を行うことができる。

また、キャラクタ 16 の表示位置を変えたい場合は、上述したオブジェクトの描画処理（図 2）で表示された画面に対して、操作部 2 での所望の操作  
5 により行われる。すなわち、その操作に対応する制御信号が操作部 2 から CPU 1 に出力され、この制御信号に基づいて CPU 1 は、変化させた表示位置に対応するカメラの行列をステップ S 11 で生成することで、以後の処理を行うことで実現できる。

そして、図 7 に表示されたエンブレム 28 のキャラクタ 16 に対する位置  
10 や大きさをユーザーが画面 15 内で確認して、これでよければ操作部 2 に所定の操作を行うことで、上述の投影の処理（図 4）がすべて行われて図 8 に示す実際のキャラクタ 16 にエンブレム 28 が貼り付いた状態で画面表示される。

このキャラクタ 16 の視点位置の移動も同様にカメラ行列又はライト行列  
15 を移動した位置に対応した行列式を生成する（ステップ S 30、ステップ S 34）ことで実現できる。

ここで、エンブレムの貼り付けについて、1 つあるいは複数のポリゴンから構成されるキャラクタ 16 の各パーツごとに行うことも可能である。

また、表示されるキャラクタ 16 は、透明に表示されるが、キャラクタ 1  
20 6 上の貼り付け位置は、最も手前、すなわちカメラからもっとも近いポリゴン面に行うようにしている。

以上のようにして、エンブレム 28 のキャラクタ 16 への投影を行うことができる。したがって、画面を見ながら貼り付けたい位置に自由にエンブレム 29 をキャラクタ 16 に貼り付けることができる。また、エンブレム 28 が  
25 貼付されたキャラクタ 16 を貼付直後に確認できるので、熟練した技能を必要とせず、だれでも所望のエンブレムを貼付することができる。さらに、操作部 2 の所定の操作により投影が行われるため、複雑な操作を必要としない。

次に、実際にゲームを行う場合に、エンブレム 28 が貼付されたキャラクタ 16 を含むゲームの表示の処理について図 9 から図 11 を用いて説明する。かかる表示処理のためのプログラムは記録媒体 4 に記録され、この記録媒体 4 をゲーム機 50 に装着されると CPU 1 の制御により主記憶装置 3 に転送され、主記憶装置 3 にて記憶され、以後の処理が行われる。もちろん、予めこの処理のためのプログラムが主記憶装置 3 に記憶されている、プログラム組み込み型のゲーム機でも適用される。

まず、ゲーム表示の処理が開始される（ステップ S 50）と、CPU 1 は、ライト行列を生成する（ステップ S 51）。上述した投影処理（図 4）のライト行列の生成（ステップ S 31）と同様である。すなわち、グローバル座標系での所定の位置（ライトの位置）を行列式で表現したものを生成する。同様に m 行 n 列で表現できる。生成したライト行列は主記憶装置 3 に出力され、一時記憶される。

次いで、CPU 1 は、投影行列を生成する処理を行う（ステップ S 52）。上述の投影処理（図 4）の投影行列の生成（ステップ S 32）と同様である。すなわち、図 6 で参照したようにライトの視点位置 30 からエンブレムテクスチャ面 31 までの距離が一定であるため、エンブレムテクスチャ面 31 のグローバル座標系の位置、テクスチャ面 31 の大きさなどを行列式で表現したものを生成する。生成した投影行列は主記憶装置 3 に出力され、一時記憶される。

次いで、CPU 1 は、ビュー行列を生成する処理を行う（ステップ S 53）。すなわち、上述のステップ S 51 及び S 52 で生成したライト行列と投影行列とを主記憶装置 3 から読み出してこれらを乗算してビュー行列を生成する。生成したビュー行列は主記憶装置 2 に出力され、一時記憶される。

次いで、CPU 1 は、デプス行列を生成する（ステップ S 54）。これは、予め主記憶装置 3 に記憶されたデプス用の行列とステップ S 51 で生成したライト行列とを乗算して生成する。生成したデプス行列は、主記憶装置 3 に

出力され一時記憶される。

次いで、CPU 1 は、現在のカメラの位置から投影行列を生成する（ステップ S 5 5）。生成した投影行列は、主記憶装置 3 に出力され一時記憶される。また、CPU 1 は、現在のカメラ位置からカメラ行列を生成する。生成したカメラ行列は主記憶装置 3 に出力され一時記憶される。

次いで、CPU 1 は、描画開始のリクエストをジオメトリックプロセッサ 5 に出力する（ステップ S 5 6）。これにより、ジオメトリックプロセッサ 5 は、描画のための処理の準備に移行する。ここでは、いわゆるプリレンダリングの処理のための描画開始のリクエストである。

10 次いで、CPU 1 は、ステップ S 5 5 で生成したカメラ行列を主記憶装置 3 から読み出す（ステップ S 5 7）。

次いで、CPU 1 は、オブジェクト（モデル）を表示したい位置に移動させる（ステップ S 5 8）。具体的には、オブジェクトの描画の処理（図 2 参照）のステップ S 1 4 と同様に、ともにグローバル座標系で表されたオブジェクトの位置とカメラの位置とから、カメラの位置を原点とした場合のオブジェクトの位置を演算する処理を行う。これにより、カメラの視点からのオブジェクトの相対位置、すなわち、実際にディスプレイ 9 の画面上に表示されるオブジェクトの位置を求めることができる。

20 次いで、CPU 1 は、モデルデータの転送を行う（ステップ S 5 9）。具体的には、CPU 1 は、ステップ S 5 8 で演算したオブジェクトの相対位置を示すモデルデータを主記憶装置 3 に出力され、一時記憶される。

次いで、CPU 1 は、描画終了のリクエストをジオメトリックプロセッサ 5 に出力する（ステップ S 6 0）。そして、CPU 1 は、主記憶装置 3 に記憶されたモデルデータを読み出してジオメトリックプロセッサ 5 に出力してオブジェクトの描画のための処理が行われる（ステップ S 6 0）。

次いで、CPU 1 は、図 1 0 のステップ S 6 2 に移行して、描画結果の Z 値をテクスチャバッファ 8 に出力させるようジオメトリックプロセッサ 5 に

指示データを出力する。ここでディスプレイ 9 の画面上にオブジェクトが表示されている場合に、画面上の任意の点（カメラの位置からの相対的なオブジェクトの位置）は 2 次元座標で表されているものの、グローバル座標系では 3 次元座標として表されている。そこで、画面上の任意の画素ごとにグローバル座標系でオブジェクトが表された場合の Z 値、すなわち奥行きを表す値をテクスチャバッファ 8 に記憶されるよう、CPU 1 は指示データを出力する。この指示データを受けたジオメトリックプロセッサ 5 は、オブジェクトの描画のための演算処理を行っているときに、各 Z 値を演算してディスプレイプロセッサ 7 にその Z 値を出力する。Z 値はディスプレイプロセッサ 7 からディスプレイ 9 を経由してテクスチャバッファ 8 に出力される。そして、テクスチャバッファ 8 において Z 値が保存されることになる。

次いで、CPU 1 は、レンダリングのため描画開始のリクエストをジオメトリックプロセッサ 5 に出力する（ステップ S 6 3）。

次いで、CPU 1 は、主記憶装置 3 に記憶したカメラ行列を読み出し（ステップ S 6 4）、カメラの位置からの相対的なオブジェクトの位置を演算する（ステップ S 6 5）。具体的には、ステップ S 5 8 と同様とともにグローバル座標系で表されたオブジェクトの位置とカメラの位置とから、カメラの位置を原点とした場合のオブジェクトの位置を演算する処理を行う。そして、CPU 1 は、演算したモデルデータを主記憶装置 3 に出力し（ステップ S 6 6）、モデルデータは主記憶装置 3 に記憶される。

次いで、CPU 1 は、ステップ S 5 4 で生成したデプス行列を主記憶装置 3 から読み出す（ステップ S 6 7）。

次いで、CPU 1 は、主記憶装置 3 に予め記憶された勾配テクスチャとステップ S 5 4 で生成したデプス行列とを主記憶装置 3 から読み出して、デプス値を生成する。ここでデプス値とは、投影の処理（図 4 参照）のステップ S 3 9 で生成した  $\alpha$  値が所定の値（たとえば“0”）である透明なモデルに対する奥行きを示す値である。生成したデプス値は主記憶装置 3 に出力され

一時記憶される。

次いで、CPU 1 は、ステップ S 5 3 で生成したビュー行列とステップ S 6 2 で生成した Z 値とをそれぞれ主記憶装置 3 とテクスチャバッファ 8 から読み出す（ステップ S 6 9）。

- 5 次いで、CPU 1 は、ステップ S 6 8 で生成したデプス値とステップ S 6 9 で読み出した Z 値とを比較する（ステップ S 7 0）。デプス値は、演算により生成したモデルの奥行きを示す値で、Z 値は実際にディスプレイ表示されたモデルの奥行きを示す値である。これらを比較することで、画面から見てどちらが手前に位置しているかを判断する。例えば、Z 値の方が大きければ白、そうでなければ黒を透明なモデルに描画する。

- 10 次いで、CPU 1 は、主記憶装置 3 に記憶されたエンブレムのデータがテクスチャバッファ 8 にコピーされるよう主記憶装置 3 とテクスチャバッファ 8 を制御する（ステップ S 7 1）。したがって、ユーザーが作成したエンブレムの表示のためのモデルデータは主記憶装置 3 とテクスチャバッファ 8 に記憶されることになる。

- 20 次いで、CPU 1 は、エンブレムテクスチャ座標の生成を行う（ステップ S 7 2）。この生成も投影の処理（図 4 参照）と同様に、作成したエンブレムがオブジェクトの各ポリゴン面に貼付されるように、エンブレムの座標を変換する。このテクスチャ座標の生成は、ステップ S 5 3 で生成したビュー行列とオブジェクトの 3 次元座標とを用いて生成する。エンブレムはテクスチャ面 3 1 では、2 次元座標（いわゆる u-v 座標系）であるが、オブジェクトのポリゴンを形成する各頂点座標は 3 次元座標であるため、生成するテクスチャ座標は 3 次元座標として表現される。かかる、テクスチャ座標は、オブジェクトの座標に変換された座標であるので、ユーザーの作成したエンブレムがキャラクタ上に貼付された画像が表示できるようになる。

次いで、CPU 1 は、フェードアウト用行列の生成を行う（ステップ S 7 3）。フェードアウト用の行列とは、エンブレムがキャラクタ上に貼付され

た場合に、例えばライトの位置によってはエンブレムが一定の直線や曲線にならずに、のこぎりの刃のように鋭角的な形状が繰り返される場合がある。

そこで、かかる形状をオブジェクト上で目立たなくする処理をおこなうためにこの行列を生成する。このフェードアウト用の行列生成のためのデータは

- 5 予め主記憶装置 3 に記憶されており、これを CPU 1 が読み出すことでこの行列を生成する。

次いで、CPU 1 は、図 11 にステップ S 7 4 に移行して、フェードアウト用行列からテクスチャ座標を生成する。具体的には、上述のステップ S 7 3 で生成したフェードアウト用行列と各オブジェクトのポリゴン面の法線ベ

- 10 クトルとからフェードアウト用の座標を生成する。

次いで、CPU 1 は、バックフェースのテクスチャを主記憶装置 3 から読み出す（ステップ S 7 5）。バックフェースとは、オブジェクトのうちディスプレイ 9 の画面上からは見えない部分のオブジェクトである。かかるバックフェースのテクスチャは、予め主記憶装置 3 に記憶されている。

- 15 次いで、CPU 1 は、色消しの処理を行う（ステップ S 7 6）。上述のステップ S 7 5 で読み出したバックフェーステクスチャに対して  $\alpha$  値を所定値にしたり、黒にしたりすることで色消しを行う。

次いで、CPU 1 は、透明モデルに対するカメラ位置を演算するため、主記憶装置 3 に記憶されたカメラの行列を読み出す（ステップ S 7 7）。そし

- 20 て、CPU 1 は、読み出したカメラの行列をもとにカメラ位置からの相対的なオブジェクトの位置を演算する（ステップ S 7 8）。演算したオブジェクトの位置は一旦主記憶装置 3 に出力して一時記憶させる（ステップ S 7 9）。

次いで、CPU 1 は、1 フレーム分の処理が終了すると描画終了のリクエストをジオメトリックプロセッサ 5 に出力する（ステップ S 8 0 で “Y e s

- 25 ” のとき）。まだ 1 フレーム分の処理が終了していないとき（ステップ S 8 0 で “N o” のとき）は、処理は再びステップ S 6 3 に移行して上述の処理を繰り返す。



描画終了のリクエストを出力した場合には、CPU 1 は、ステップ S 8 1 に処理が移行し、オブジェクトの描画の処理を行う（ステップ S 8 1）。具体的には、CPU 1 は、ステップ S 5 9、ステップ S 6 6、及びステップ S 7 9 で主記憶装置 3 に記憶されたモデルデータを読み出しジオメトリックプロセッサ 5 に出力し、実際にエンブレムが貼付されたオブジェクトを含む画面全体の描画をするための演算を行う。

演算したデータは、ディスプレイプロセッサ 7 からディスプレイ 9 に出力され、実際にディスプレイ 9 に表示されることになる。これにより、実際にゲーム実行中、エンブレム 2 8 が貼付されたキャラクタ 1 6 を含む、1 フレームの画像を表示することができる。そして、上述した処理を繰り返すことで連続した画面表示を行うことができる。

ディスプレイに表示された例を図 1 2 に示す。画面 1 5 内にはユーザーが操作を行うキャラクタ（オブジェクト）1 6 が位置し、上述したエンブレム 2 8 が、キャラクタ 1 6 の背面に実際に貼り付けられたように位置している。

次に、ユーザーによって作成したエンブレム画像がオブジェクト 1 6 の各面の表面、裏面にすべて貼り付けられ、これを視点から最も近い投影面にのみエンブレムを貼り付けることに関して、図 1 3 乃至図 1 9 を参照して説明する。

この処理は上述したゲーム実行中の表示処理（図 9 乃至図 1 1 参照）でステップ S 7 2、S 7 3 の段階で、CPU 1 の制御信号を受けたジオメトリックプロセッサ 5 によって実行される（図 1 0 参照）。

まず、図 1 3 A に示すようにエンブレム “A” という文字が例えば図 5 のエンブレム作成画面で作成した場合で説明する。

ここで、3 次元空間座標系（グローバル座標系）の任意の位置座標を（ $x$ 、 $y$ 、 $z$ 、1）、エンブレムが存在するテクスチャ空間上の座標を（ $s$ 、 $t$ 、1）としたときに、この 3 次元座標系の位置（ $x$ 、 $y$ 、 $z$ 、1）をテクスチャ空間上の座標（ $s$ 、 $t$ 、1）へマッピング（テクスチャマッピング）する

ための行列式は以下のように表される。

[数式 1]

$$\begin{bmatrix} 2 * scaleS / (r - l) & 0.0 & -scaleS * (r + l)(r - l) + transS & 0.0 \\ 0.0 & 2 * scaleT / (t - b) & -scaleT * (t + b)(t - b) + transT & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \dots$$

(1)

- 5      ここで、行列式中の“*t*”、“*b*”、“*l*”、“*r*”は、テクスチャ面 3 1 の大きさを表すためのパラメータで、それぞれテクスチャ空間における原点からの上方向の長さ (top)、下方向の長さ (bottom)、右方向の長さ (left)、左方向の長さ (right) を示す。この関係を図 1 4 に示す。ここで、図 1 5 A は、エンブレムとオブジェクト 1 6 との関係を表した図面である。オブジェクト 1 6 に対してテクスチャ面 3 1 が図のように存在し、その面には予め作成したエンブレム “A” が位置する。ここでテクスチャ面 3 1 の大きさを表す 4 つのパラメータは、テクスチャ空間上における面 3 1 の大きさを表すものである。

- 15      また、“*scaleS*”、“*scaleT*”は、テクスチャ面 3 1 で作成したエンブレムをオブジェクト 1 6 の位置する 3 次元空間座標系に位置させたときのエンブレムの大きさを調整するためのパラメータである。それぞれ横方向、縦方向の大きさを調整するものである。これにより、テクスチャ面 3 1 で作成したエンブレムの大きさを 3 次元空間座標系内で調整することができる。図 1 5 A で説明すると、このエンブレム “A” をオブジェクト 1 6 に貼り付けたとき  
20      の 3 次元座標空間上のエンブレムの大きさを調整することができる。

- さらに、“*transS*”、“*transT*”は、テクスチャ面 3 1 の原点と 3 次元空間座標系の原点との位置ずれを補正するためのパラメータである。それぞれ、横方向と縦方向を表す。テクスチャ面 3 1 の原点は例えば図 1 4 の (0、0) の位置であって、3 次元空間座標系の原点位置とは必ずしも一致しない。図  
25      1 5 B は、図 1 5 A に対してオブジェクト 1 6 の先端を正面としたときの断

面図を示す。図 1 5 Aにおいてテクスチャ面 3 1 の中心位置をテクスチャ空間上の原点にとったとき、3 次元空間座標系の原点がこの位置にないと、3 次元空間座標系にこのエンブレムを貼り付けたときに、図 1 5 Bの面 A（オブジェクト 1 6 の羽の部分）にエンブレムが位置することができなくなってしまう。この 2 つのパラメータを調整することで、両座標系の原点位置のずれを補正することができる。

この行列式 (1) において、例えば、 $t = -1.0$ 、 $b = 1.0$ 、 $l = -1.0$ 、 $r = 1.0$ （テクスチャ空間上のテクスチャ面の大きさが原点から“1”の大きさ）、 $scaleS = 0.5$ 、 $scaleT = 0.5$ （3 次元空間座標系でのテクスチャ面の大きさはテクスチャ空間座標系の半分）、 $transS = 0.5$ 、 $transT = 0.5$ （テクスチャ面の原点位置を縦、横 0.5 ずつ、ずらした位置が 3 次元空間座標系での原点）としたとき、行列式 (1) は、

[数式 2]

$$\begin{bmatrix} 0.5 & 0.0 & 0.5 & 0.0 \\ 0.0 & 0.5 & 0.5 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \quad \dots (2)$$

と表される。この行列式 (2) から 3 次元空間座標系の任意の位置 ( $x$ 、 $y$ 、 $z$ 、1) に対するテクスチャ空間座標系の任意の位置 ( $s$ 、 $t$ 、1、1) は、

[数式 3]

$$s = 0.5 * x + 0.5, \quad t = 0.5 * y + 0.5 \quad \dots (3)$$

となる。この 2 つの式 (3) は、 $x$ 、 $y$  が等しい 3 次元空間座標系の点には同一のテクスチャ座標がマッピングされることを示す。すなわち、3 次元空間座標系で  $x$ 、 $y$  の値が等しい点において、この 3 次元空間座標系で奥行きを示す  $z$  の値がどのような値をとっても、テクスチャ座標の同一の点が 3 次元座標系に位置することになる。このことを、図面を用いて説明すると、図 1 5 Bに示すようにオブジェクト 1 6 は、各パーツに対して面 A から面 H が存在する。オブジェクト 1 6 の翼の部分には 2 つの面 A、面 B から構成され

る。このとき、奥行き  $z$  の値をどのようにとっても  $x$ 、 $y$  の位置が等しい  
 と同様のテクスチャ面がマッピングされることになるから、図 16 A に示す  
 ようにテクスチャ面 31 に作成したエンブレム “A” は、オブジェクト 16  
 の面 A のみならず、その奥行き ( $z$  方向) に向けて、面 B、面 C、面 D、面  
 5 F、面 F、面 G、面 H すべての面に貼り付けられることになる (図 16 B も  
 参照)。

本発明では、このようにすべての面に貼り付けられたエンブレムを、視点  
 から最も近い面にのみ貼り付けるようにジオメトリックプロセッサ 5 (図 1  
 参照) において処理を行う。まず、3次元空間上の法線 ( $x$ 、 $y$ 、 $z$ 、1)  
 10 をテクスチャ空間 ( $s$ 、 $t$ 、1、1) にマッピングするための以下のような  
 行列式を考える。

[数式 4]

$$\begin{bmatrix} 0.0 & 0.0 & 0.5 & 0.5 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \cdots (4)$$

このとき、 $s$ 、 $t$  は、

15 [数式 5]

$$s = 0.5 * z + 0.5, t = 0 \cdots (5)$$

を得る。すなわち、テクスチャ空間上の横方向の任意の点  $s$  は、どのような  
 $t$  の値をとっても  $t$  には依存せず、また、3次元空間座標系で  $z$  成分が “1”  
 (奥行き) のとき、 $s = 1.0$ 、 $z$  成分が “-1” のとき  $s = 0.0$  となる。つまり、  
 20 法線が視点方向のとき  $s$  の値は、“1. 0”、視点と反対方向のとき  $s$  の値は  
 “0. 0” となる値をテクスチャブレンドの  $\alpha$  値 (透明度) とすること  
 で、視点と反対方向の面 (図 16 B において、面 B、面 D、面 F、面 H)  
 に貼り付けられたエンブレムの透明度を “0” にするのである。

図 17 を用いて説明すると、この図 17 はテクスチャ空間上の任意の点を  
 25 示し、式 (5) に示すように  $s$  の値は  $t$  の値に依存せず、各テクスチャ空間

上の任意の点が $\alpha$ 値を取るようなエンブレムを作成したテクスチャ面とは異なるテクスチャを用意する。例えば、図1の主記憶装置2に予め格納されていてもよいし、記録媒体4に格納されていてもよい。このテクスチャは、 $s$ の値が小さければ小さいほど $\alpha$ 値が0に近づくよう、つまり、 $s$ の値が小さいほど徐々に透明になるように構成している。よって、 $s$ の値が“0”のとき透明度を“0”(見えなくする)にすることで、オブジェクト16の背面(面B、面D、面F、面H)に貼り付けられたエンブレムを透明にするのである。

次に、ジオメトリックプロセッサ5は、オブジェクト16の面A、面C、面E、面Gに貼り付けられたエンブレムのうち、視点から最も近い面Aについてのみエンブレムを残すようにする。そのために、ジオメトリックプロセッサ5は各面に対する $\alpha$ 値を比較することで、最も視点から近い面を判断する。従って、図19A、Bに示すようにエンブレム“A”がオブジェクト16の視点から最も近い羽の部分に貼り付けられることになる。

なお、この $\alpha$ 値の比較であるが、オブジェクト16を構成する画素ごとに比較することで最も近い面を判断している。すなわち、図19Aに示すようにテクスチャ面31に作成したエンブレム“B”がオブジェクト16の羽の部分からはみ出して位置している場合で考える。この場合も、図20A、Bに示すように各面(面Aから面H)にエンブレム“B”が張り付いて、これを背面(面B、面D、面F、面H)に位置するエンブレムに透明度“0”のテクスチャをマッピングさせて見えないようにし、 $\alpha$ 値によって最も近い面にエンブレム“B”が貼り付いている。しかし、図21A、Bに示すようにこの $\alpha$ 値を各画素で比較すると、エンブレム“B”がはみ出して貼り付いているオブジェクト16の胴体部分は、テクスチャ面31の上部の部分から画素単位に $\alpha$ 値を比較すると最も近い位置に位置することになる。よって、はみ出したエンブレムは、視点から最も近いオブジェクト16の位置に貼り付けられることになる。

請求の範囲；

1. 画面内で所定の操作を行うための操作手段と、所定の演算を行う演算処理手段と、前記操作手段と前記演算処理手段とに接続され前記演算処理手段
- 5 を制御する制御手段とを有するゲーム機において、ゲームのキャラクタに所定の画像を投影するプログラムであって、

前記操作手段の操作により2次元座標からなる画像を前記制御手段により生成する画像作成ステップと、

- 前記操作手段からの入力信号に基づいて、前記画像作成ステップで作成し
- 10 た画像をキャラクタに投影するための仮想光源と、前記画像とを、三次元仮想空間内において前記キャラクタ周囲の任意の位置に配置し、前記投影光源から前記キャラクタに前記画像を投影した投影像を前記キャラクタに貼付するステップと、

を実現させるためのプログラム。

15

2. 前記キャラクタは、複数のパーツの組み合わせによって構成されており、前記制御手段に、前記操作手段の操作に応じて前記パーツのうち少なくとも一つが前記画像の投影対象として指定させ、前記指定されたパーツに対して前記投影画像を貼り付けさせる、
- 20 ことを実現させるための請求項1記載のプログラム。

3. 画面内で所定の操作を行うための操作手段と、所定の演算を行う演算処理手段と、前記操作手段と前記演算処理手段とに接続され前記演算処理手段を制御する制御手段とを有し、ゲームのキャラクタに所定の画像を投影する
- 25 プログラムが組み込まれたゲーム機であって、

前記操作手段の操作により2次元座標からなる画像を前記制御手段により生成する画像作成ステップと、

前記操作手段からの入力信号に基づいて、前記画像作成ステップで作成した画像をキャラクタに投影するための仮想光源と、前記画像とを、三次元仮想空間内において前記キャラクタ周囲の任意の位置に配置し、前記投影光源から前記キャラクタに前記画像を投影した投影像を前記キャラクタに貼付する

5 ステップと、

を実現させるためのプログラムが組み込まれたゲーム機。

4. 前記キャラクタは、複数のパーツの組み合わせによって構成されており、前記制御手段に、前記操作手段の操作に応じて前記パーツのうち少なくとも一つが前記画像の投影対象として指定させ、前記指定されたパーツに対して前記投影画像を貼り付けさせる、

10 ことを実現させるための請求項3記載のプログラムが組み込まれたゲーム機。

15 5. 画面内で所定の操作を行うための操作手段と、所定の演算を行う演算処理手段と、前記操作手段と前記演算処理手段とに接続され前記演算処理手段を制御する制御手段とを有するゲーム機に対してゲームのキャラクタに所定の画像を投影するプログラムが格納された記録媒体であって、

前記操作手段の操作により2次元座標からなる画像を前記制御手段により  
20 生成する画像作成ステップと、

前記操作手段からの入力信号に基づいて、前記画像作成ステップで作成した画像をキャラクタに投影するための仮想光源と、前記画像とを、三次元仮想空間内において前記キャラクタ周囲の任意の位置に配置し、前記投影光源から前記キャラクタに前記画像を投影した投影像を前記キャラクタに貼付する

25 ステップと、

を実現させるためのプログラムが格納された記録媒体。

6. 前記キャラクタは、複数のパーツの組み合わせによって構成されており、前記制御手段に、前記操作手段の操作に応じて前記パーツのうち少なくとも一つが前記画像の投影対象として指定させ、前記指定されたパーツに対して前記投影画像を貼り付けさせる、

5      ことを実現させるための請求項 5 記載のプログラムが格納された記録媒体。

7. 三次元仮想空間内において、三次元座標からなるオブジェクトに所定の投影画像を投影したオブジェクト画像をディスプレイ装置に表示するための画像データを生成する画像データ生成手段としてコンピュータを動作させる

10   画像表示制御プログラムであって、

前記画像表示制御プログラムは、制御手段に、

前記オブジェクトに投影される所定の投影画像データを生成させ、

前記仮想空間内に前記オブジェクトおよび前記投影画像を配置させ、

操作者の操作に基づいて前記オブジェクトに対する前記投影画像の相対位

15   置と、前記投影画像を前記オブジェクトに投影する仮想光源の位置を決定させ、

前記仮想光源と前記オブジェクト上の投影位置を含む投影面の距離を演算し、前記仮想光源から所定距離以上はなれた投影面を投影対象から除いて、前記オブジェクトにおいて前記投影画像を投影すべき投影面を演算させ、

20   前記仮想光源を視点として前記投影画像を前記投影面に投影して、前記投影画像を前記オブジェクトの前記投影面に貼り付けさせ、

前記投影画像が貼り付けられた前記オブジェクトについてのオブジェクト画像データを生成させる機能を実現させることを特徴とする画像表示制御プログラム。

25

8. 前記制御手段に、さらに前記投影画像の画像データをユーザーが操作手段から入力する操作信号に基づいて生成させる機能を実現させることを特徴



とする請求項 7 記載の画像表示制御プログラム。

9. 前記制御手段に、前記投影画像を前記オブジェクトと同一ないし近似する形状を備える透明オブジェクトに対して投影して、前記投影画像を前記透

5 明オブジェクトに貼り付けさせ、

前記表示装置に前記投影画像を貼り付けた透明オブジェクトを前記オブジェクトに重ねて表示させる機能を実現させることを特徴とする請求項 7 又は 8 記載の画像表示制御プログラム。

10 10. 前記仮想光源と前記投影オブジェクトと前記オブジェクトの位置関係によって、前記オブジェクトに貼り付けられた投影画像の変形が著しいと判断される場合、前記投影オブジェクトの透明度を変化させる機能を実現させることを特徴とする請求項 7 乃至 9 の画像表示制御プログラム。

15 11. 前記オブジェクトが複数のパーツの組み合わせによって構成されており、

前記制御部に、操作者の操作に応じて前記パーツのうち少なくとも一つが前記投影画像の投影対象として指定させ、前記指定されたパーツに対して前記投影画像を貼り付けさせる機能を実現させることを特徴とする請求項 7 乃

20 至 10 記載の画像表示制御プログラム。

12. 三次元仮想空間内において、三次元座標からなるオブジェクトに所定の投影画像を投影したオブジェクト画像をディスプレイ装置に表示するための画像データを生成する画像データ生成手段としてコンピュータを動作させ

25 る画像表示制御プログラムであって、

前記画像表示制御プログラムは、制御手段に、

前記オブジェクトに投影される所定の投影画像データを生成させ、

前記仮想空間内に前記オブジェクトおよび前記投影画像を配置させ、

操作者の操作に基づいて前記オブジェクトに対する前記投影画像の相対位置と、前記投影画像を前記オブジェクトに投影する仮想光源の位置を決定させ、

- 5 前記仮想光源と前記オブジェクト上の投影位置を含む投影面の距離を画素単位で演算し、前記仮想光源を視点として前記投影画像の画素を前記視点から最も近い前記投影面上の画素に投影して、前記投影画像を前記オブジェクトの前記投影面に貼り付けさせ、

前記投影画像が貼り付けられた前記オブジェクトについてのオブジェクト

- 10 画像データを生成させる機能を実現させることを特徴とする画像表示制御プログラム。

## 要約

- ユーザーが作成した表示を、複雑な操作をせずに、キャラクタが複雑な形でも自由にそのキャラクタに貼り付けることのできるプログラムを提供する。画面内で所定の操作を行うための操作手段と、所定の演算を行う演算処理手段と、前記操作手段と前記演算処理手段とに接続され前記演算処理手段を制御する制御手段とを有するゲーム機において、ゲームのキャラクタに所定の画像を投影するプログラムであって、前記操作手段の操作により2次元座標からなる画像を前記制御手段により生成する画像作成ステップと、前記操作手段からの入力信号に基づいて、前記画像作成ステップで作成した画像をキャラクタに投影するための仮想光源と、前記画像とを、三次元仮想空間内において前記キャラクタ周囲の任意の位置に配置し、前記投影光源から前記キャラクタに前記画像を投影した投影像を前記キャラクタに貼付するステップと、を実現させるためのプログラム。